REGION AND OBJECT SEGMENTATION ALGORITHMS IN THE QIMERA SEGMENTATION PLATFORM

Noel O'Connor[1], Sorin Sav[1], Tomasz Adamek[1], Vasileios Mezaris[2], Ioannis Kompatsiaris[2], Tsz Ying Lui[3], Ebroul Izquierdo[3], Christian Ferran Bennström[4], Josep R Casas[4]

[1]Centre for Digital Video Processing, Dublin City University, Ireland
[2]Informatics and Telematics Institute, 1st Km Thermi-Panorama Rd,, Thessaloniki 57001, Greece
[3]Department of Electronic Engineering, University of Queen Mary, Mild End Road, London, U.K.
[4] Signal Theory & Communications Department, Technical University of Catalonia, Spain

ABSTRACT

In this paper we present the Qimera segmentation platform and describe the different approaches to segmentation that have been implemented in the system to date. Analysis techniques have been implemented for both region-based and object-based segmentation. The region-based segmentation algorithms include: a colour segmentation algorithm based on a modified Recursive Shortest Spanning Tree (RSST) approach, an implementation of a colour image segmentation algorithm based on the K-Means-with-Connectivity-Constraint (KMCC) algorithm and an approach based on the Expectation Maximization (EM) algorithm applied in a 6D colour/texture space. A semi-automatic approach to object segmentation that uses the modified RSST approach is outlined. An automatic object segmentation approach via snake propagation within a level-set framework is also described. Illustrative segmentation results are presented in all cases. Plans for future research within the Qimera project are also discussed.

1. INTRODUCTION

The Qimera initiative is a pan-European collaborative voluntary research project. Its objective is to develop a flexible modular software architecture for video object segmentation and tracking which can be used as a vehicle and test-bed for collaborative algorithm development. The background to the project and the structure of the very first version of the system developed is described in [1]. Since the version described in [1], a number of analysis tools, developed by Qimera partners, have been integrated into the system. In this paper, we describe each analysis technique integrated to date and present some illustrative results in each case. A comparison of the performance of the different techniques is not presented here, since this is the subject of an ongoing work item within the project and will be reported upon in a subsequent publication. The analysis techniques that have been developed can be loosely categorised as either region-based or object-based approaches to segmentation. The region-based approaches are described in Section 2 of this paper. The object-based approaches are described in Section 3. The future work planned within Qimera is briefly outlined in Section 4.

2. REGION-BASED SEGMENTATION TOOLS

2.1. Modified RSST

This approach is based on a straightforward extension of the well known Recursive Shortest Spanning Tree (RSST) algorithm [2]. The algorithm is explained in detail in [1] and is only briefly described here. We modify the original algorithm to avoid merging regions with very different colours. To this end, we carry out the RSST in the HSV colour space and add a second merging stage in which we do not penalize large regions.

2.1.1. Results

Illustrative results of the modified algorithm versus the standard RSST for images from the Foreman and Table Tennis sequences are presented in Figure 1.

2.2. Region-based segmentation via KMCC

The segmentation scheme presented in this section is based on combining a novel segmentation algorithm and a procedure for accelerating its execution. The segmentation algorithm is based on a variant of the K-Means-with-connectivity-constraint algorithm (KMCC) [3], to produce connected regions. The acceleration procedure is based on using properly reduced versions of the original images as input and performing partial pixel reclassification after the convergence of the KMCC-based algorithm. The overview of the proposed scheme is presented in Fig. 2.

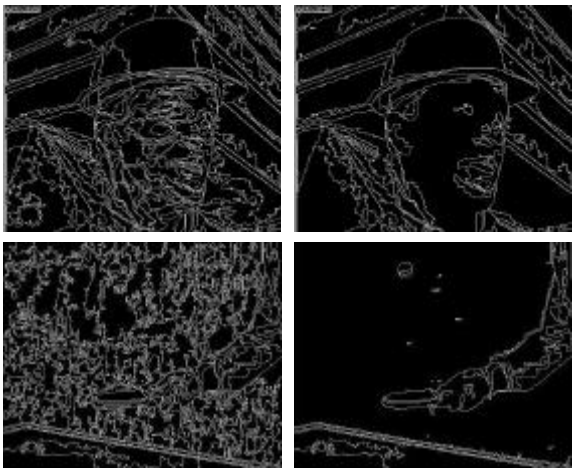


Figure 1: RSST (left) vs. modified RSST (right)

The KMCC-based segmentation algorithm is made of four steps. The first step is the estimation of the color and texture features to be used for pixel classification. The color feature vector of pixel $\boldsymbol{p} = [p_x \; p_y]$ is made of the three intensity coordinates of the CIE L*a*b* color space: $\boldsymbol{I}(\boldsymbol{p}) = [I_L(\boldsymbol{p}) I_a(\boldsymbol{p}) I_b(\boldsymbol{p})]$. In order to calculate a texture feature vector $\boldsymbol{T}(\boldsymbol{p})$ for pixel $\boldsymbol{p}$, the 2D fast iterative scheme for performing the Discrete Wavelet Frames (DWF) decomposition is used, based on the low-pass Haar filter.

The feature estimation is followed by an initial clustering procedure, in order to compute the initial values required by the KMCC algorithm. For that, the image is broken down into square non-overlapping blocks of dimension f and intensity and texture feature vectors are assigned to each block. The number of regions of the image is initially estimated by applying a variant of the *maximin* algorithm to the blocks, followed by the application of a simple K-Means algorithm. Using the output of the K-Means and a recursive component-labeling algorithm, a total of K' connected regions s_k are identified. Their intensity, texture and spatial centers, $\bar{\boldsymbol{I}}_k$, $\bar{\boldsymbol{T}}_k$ and $\bar{\boldsymbol{S}}_k$ respectively, are then calculated as the mean values of the features of the pixels belonging to the blocks assigned to each region.

In the conditional filtering step, a moving average filter alters the intensity features in those parts of the image where intensity fluctuations are particularly pronounced. The decision of whether the filter should be applied to a particular pixel $\boldsymbol{p}$ is made by evaluating the norm of its texture feature vector $\boldsymbol{T}(\boldsymbol{p})$; the filter is not applied if that norm is below a threshold T_{th}. The output of the conditional filtering module can be expressed as:

$$\boldsymbol{J}(\boldsymbol{p}) = \begin{cases} \boldsymbol{I}(\boldsymbol{p}), & \text{if } \|\boldsymbol{T}(\boldsymbol{p})\| < T_{th} \\ \dfrac{1}{f^2}\displaystyle\sum_{m=1}^{f^2} \boldsymbol{I}(\boldsymbol{p}_m), & \text{if } \|\boldsymbol{T}(\boldsymbol{p})\| \geq T_{th} \end{cases}$$

$$T_{th} = \max\{0.65 \cdot T_{\max}, 14\}$$

where $T_{\max}$ is the maximum value of the norm $\|\boldsymbol{T}(\boldsymbol{p})\|$ in the image. Region intensity centers calculated using these filtered intensities are denoted $\bar{\boldsymbol{J}}_k$.

In the final step of the proposed segmentation algorithm, the pixels are classified into regions by a variant of the KMCC algorithm, using a distance of a pixel $\boldsymbol{p}$ from a region s_k defined as:

$$D(\boldsymbol{p}, s_k) = \|\boldsymbol{J}(\boldsymbol{p}) - \bar{\boldsymbol{J}}_k\| + \|\boldsymbol{T}(\boldsymbol{p}) - \bar{\boldsymbol{T}}_k\| + l\,\frac{\bar{A}}{A_k}\|\boldsymbol{p} - \bar{\boldsymbol{S}}_k\|$$

where $\|\boldsymbol{J}(\boldsymbol{p}) - \bar{\boldsymbol{J}}_k\|$, $\|\boldsymbol{T}(\boldsymbol{p}) - \bar{\boldsymbol{T}}_k\|$ and $\|\boldsymbol{p} - \bar{\boldsymbol{S}}_k\|$ are the Euclidean distances of the intensity, texture and spatial feature vectors respectively, A_k is the area of region s_k, $\bar{A}$ is the average area of all regions and l is a regularization parameter. The KMCC algorithm performs splitting of non-connected regions and merging of neighboring regions with similar intensity or texture centers. The region centers are recalculated in every iteration and centers corresponding to regions that fall below a size threshold $th_{size} = 0.75\%$ of the image area, are omitted.

In order to accelerate the completion of the segmentation process, the above-presented segmentation algorithm is applied to properly reduced images, where each pixel corresponds to a $R \times R$ square block of the original image. The value of R is chosen so that even undesirable regions (falling below the size threshold th_{size}) are detectible in the reduced image. This technique improves the time efficiency of the segmentation process; however, edges between objects are crudely approximated by piecewise linear segments, thus lowering the perceptual quality of the result [4]. To alleviate this problem, the subsequent reclassification of pixels belonging to blocks on edges between regions is proposed. If a block, assigned to one region, is neighboring to blocks of Γ other regions, $\Gamma \neq 0$, the assignments of all pixels of the original image represented by that block must be re-evaluated, since each of them may belong to any one of the possible $\Gamma + 1$ regions. The reclassification of these disputed pixels is performed using their intensity values in the CIE L*a*b color space and a Bayes classifier. This process is illustrated in Fig. 2.

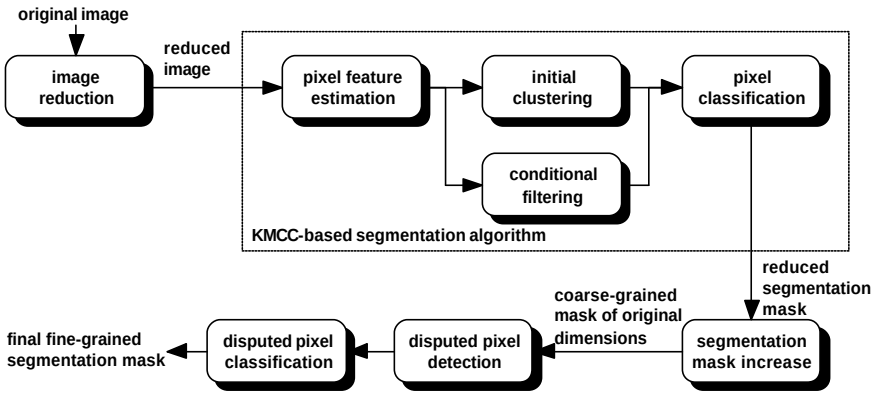


Figure 2: Overview of the KMCC-based segmentation scheme.

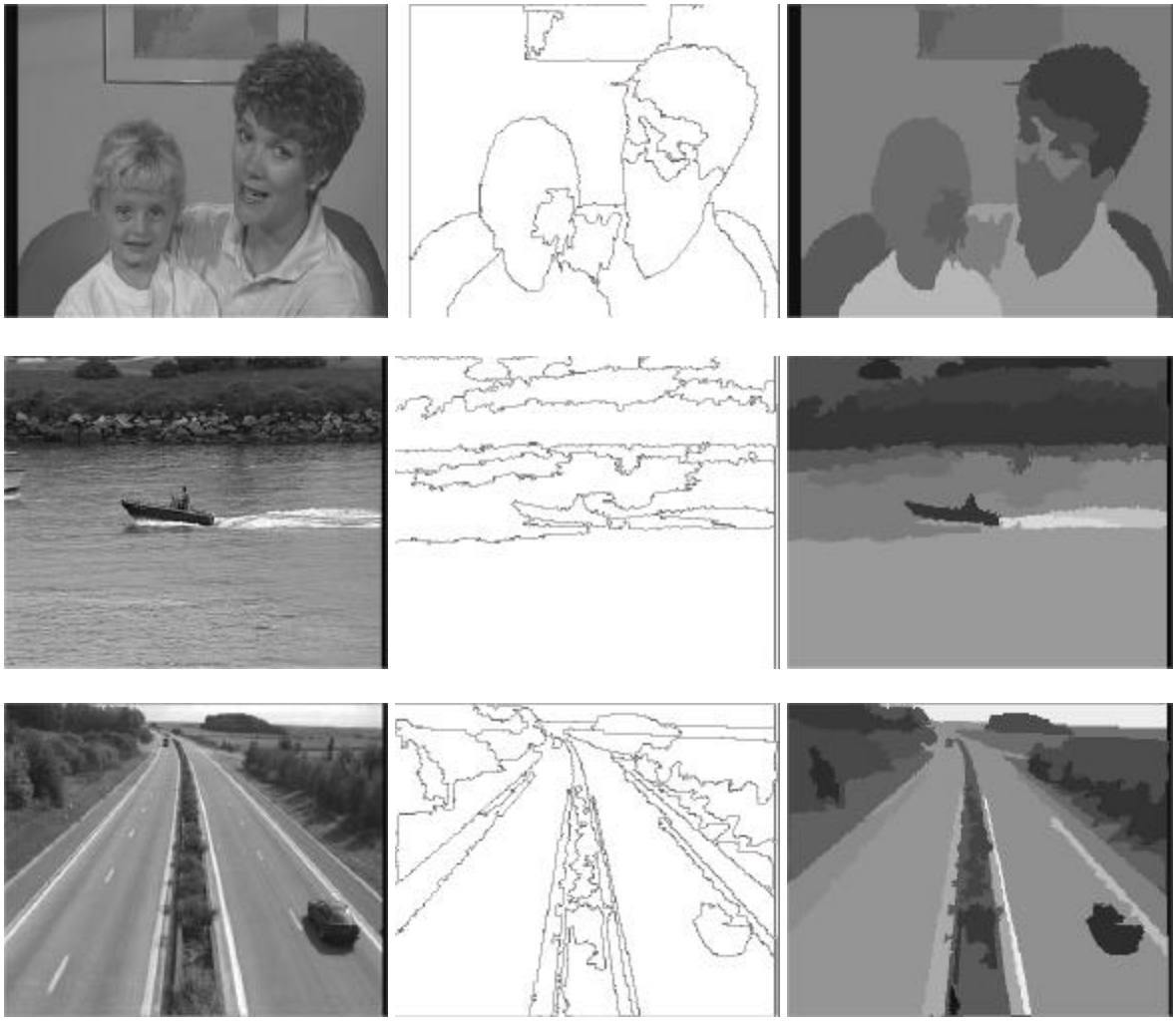


Figure 3: Results of the KMCC-based segmentation scheme on still images.

2.2.1. Results

Some results of the application of the aforementioned scheme to still images are presented in Fig. 3 (region boundaries and corresponding segmentation masks). It can be seen that the proposed segmentation scheme has succeeded is creating meaningful regions without blocky contours.

2.3. Region-based segmentation via the EM algorithm

In this approach, we assume that the spatial distribution of colour and texture primitives in the video frames can be modelled as a mixture of Gaussians. We use six features for segmentation, three colour and three texture components. The three colour components are computed from the well-known CIE L*a*b* colour space, where L* encodes luminance and a* and b* denote the range from red to green and range from yellow to blue, respectively. The other three texture components are defined in terms of anisotropy, contrast and orientation primitives extracted from second order statistics M_s of the pixels in small neighbourhoods.

The M_s for each pixel is defined as

$$M_s(x,y) = \begin{pmatrix} M_{20} & M_{11} \\ M_{11} & M_{02} \end{pmatrix}$$

and the corresponding second order moments are:

$$M_{pq} = \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}(i-x_c)^p(j-y_c)^q f(i,j)$$

where $f(i,j)$ is the image intensity at position i, j and x_c, y_c are the co-ordinates of the region's centroid, which are defined as:

$$x_c = \frac{M_{10}}{M_{00}} \text{ and } y_c = \frac{M_{01}}{M_{00}}.$$

The result of M_s is a 2x2 symmetric positive semidefinite matrix which provides three pieces of information about each pixel's orientations along the x, y and xy dimensions. We compute each pixel's anisotropy, contrast and orientation, represented as a, c and q respectively, using the following equations:

$$a = \frac{l_1 - l_2}{l_1 + l_2},\ c = 2\sqrt{l_1 + l_2},\ q = \tan^{-1}\left(\frac{l_2}{l_1}\right)$$

where l_1 and l_2 are the first and second eigenvalues of M_s. Since the matrix M_s is symmetric positive semidefinite, the eigenvalues can be computed easily from the equation

$$l_{1,2} = \frac{1}{2}(j_{11} + j_{22} \pm \sqrt{(j_{11} - j_{22})^2 + 4j_{12}^2})$$

with $l_1 \geq l_2$.

Once the features are extracted from the image, the Expectation Maximization (EM) algorithm is used to determine the maximum likelihood parameters of a mixture of K Gaussians in the feature space as described in [4]. The feature vectors are modelled as a mixture of Gaussian distributions of the form:

$$f(X_i|\Phi) = \sum_{j=1}^{k} p_j f_j(X_i|q_j)$$

where $f(X_i|q_j) = \frac{1}{(2p)^{\frac{d}{2}}|\Sigma_j|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_i-m_j)^T\Sigma_j^{-1}(X_i-m_j)}$ is the probability density function for cluster j, $q_j = (m_j, \Sigma_j)$ is the set of parameters for density function $f_j(X_i|q_j)$, m_j is the mean of cluster j, p_j is the mixing proportion of cluster j subject to the condition $p_j \geq 0$ and $\sum_{j=1}^{k} p_j = 1$ where K is the number of components. X_i is a 6-dimensional feature vector, $\Phi = (p_1, p_2 \ldots p_k, q_1, q_2 \ldots q_k)$ is the set of all parameters, and $f(X_i|\Phi)$ is the probability density function of our observed data point X_i given parameters Φ.

The maximization is performed by the following iteration.

$$E[z_{ij}] = p(z_{ij} = 1|X, \Phi^{(t)}) = \frac{p_j^{(t)} p_j(X_i|\Phi_j^{(t)})}{\sum_{s=1}^{k} p_s(X_i|\Phi_s^{(t)}) p_s^{(t)}},$$

$$p_j^{(t+1)} = \frac{1}{N}\sum_{i=1}^{N} E[z_{ij}],\ m_j^{(t+1)} = \frac{1}{Np_j^{(t+1)}}\sum_{i=1}^{N} E[z_{ij}]X_i,$$

$$\Sigma_j^{(t+1)} = \frac{1}{Np_j^{(t+1)}}\sum_{i=1}^{N} E[z_{ij}](X_i - m_j^{(t+1)})(X_i - m_j^{(t+1)})^T$$

where $E[z_{ij}]$ is the expected value of the probability that the data belongs to cluster j and $\sum_{i=1}^{N} E[z_{ij}]$ is the estimated number of data points in class j. At each iteration, the model parameters are re-estimated to maximize the model log-likelihood, $\log f(X|\Phi)$, until convergence. If the initial parameters are good approximations the iteration converges to the true solutions. We initialise the mean vectors by randomly choosing from data points with some Gaussian noises and constrain the initial covariances to be proportional to the identity matrix. Finally we choose K to be ranged from 2 to 6 and use the MDL [6][7] principle to select the optimal number of mixture components automatically.

After running the EM algorithm, each image pixel is labelled with the cluster for which it attains the highest likelihood. A 3X3 max-vote filter is then applied to smooth the image and a connected-component algorithm is performed to produce a set of homogeneous image regions.

2.3.1. Results

Some results of using the EM segmentation algorithm are illustrated in Figure 4.

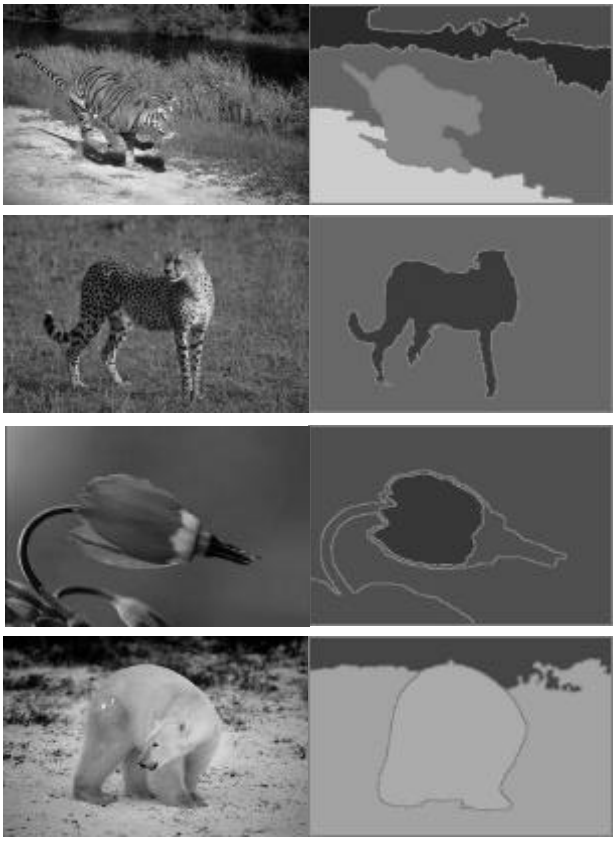
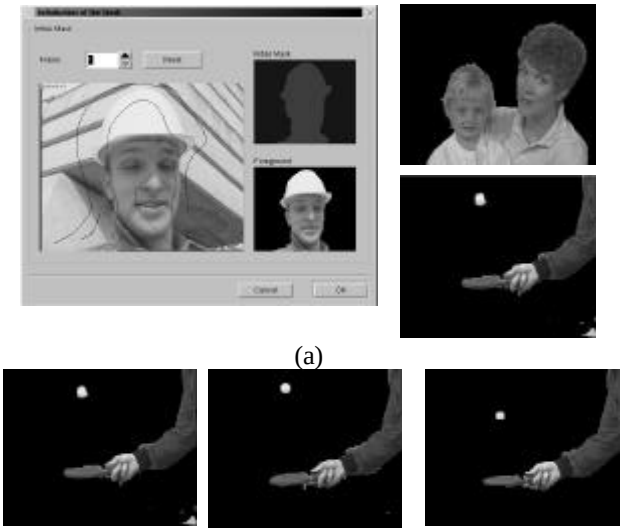


Figure 4: Segmentation results (right) on some test images (left).

3. OBJECT-BASED SEGMENTATION TOOLS

3.1. Semi-automatic object segmentation

In this approach user interaction is performed in order to define what objects are to be segmented within an initial image in the sequence [1]. The result of user interaction is then used in an automatic segmentation process. It is desirable that user interaction be easily and quickly performed. To this end, we allow the user to mark objects to be segmented via a simple mouse drag over each object. The interface provided and an example of user interaction is presented in Fig. 5(a). The results of the automatic segmentation process based on this interaction are also presented to the used within the interface. The automatic process uses the result of the algorithm described in Section 2.1. Regions coincident with an object's mouse drag are added to that object. Unclassified objects are assigned to competing objects using a normalized distance criterion similar to that used in the modified RSST algorithm. For complex objects, user interaction (and the subsequent automatic process) can be iteratively applied in order to refine the segmentation result.



(a)

(b)

Figure 5: Semi-automatic object segmentation results

3.1.1. Object Tracking

In order to track segmented objects, motion estimation is carried out and regions in the current image are back projected onto the previous segmentation mask. A set of rules are then applied in order to assign each region to an object [1].

The results of segmenting a selection of objects from well known test sequences are presented in Fig 5(a). The result of tracking objects from one sequence (Table Tennis) are presented in Fig. 5(b).

3.2. Automatic object segmentation

This approach focuses on object segmentation over video sequences. An automatic *snake* segmentation tool has been implemented within a level-set framework following the work presented in [9]. The partial differential equation (PDE) modelling the surface evolution includes a unified boundary term and a region information term.

3.2.1. Level-set based snake segmentation

Following the work of M. Kass *et al.* who introduced *snakes* as energy-based *active contours*, Caselles *et al* proved the equivalence between the *snake* model and the *geodesic active contour* formulation. In [9], Paragios *et al.* extended this concept to the *geodesic active regions.* In his approach *Paragios* unifies both region and boundary information in a level set framework. Using the work of Osher and Sethian *snakes* can be implemented as the zero level set of a higher dimensional surface j such that $\{j(C(t),t)=0\}$. This can be better visualized in Fig. 6.

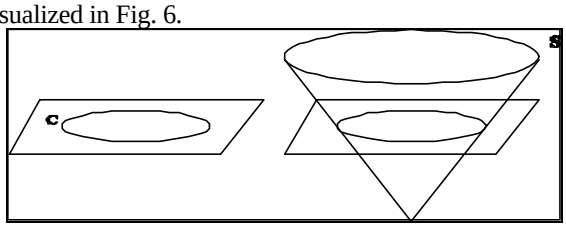


Figure 6: Level set of an embedding function j for a closed curve C in R^2.

This approach to *snake* segmentation can handle topological changes of the objects under analysis. The *snake*s at the zero level set can split in order to accommodate unconnected components of the object, while the overall connectivity is imposed by a single evolving surface. Moreover the curve representation is implicit, parameter-free and intrinsic. The surface evolution is driven by internal and external forces. Internal forces are associated to the *snake* model, whereas the external forces are related to the visual properties of the object such as contour and luminance homogeneity. The contribution of each force term can be balanced by using a coefficient a as shown in expression (1).

$$F = \alpha F_{Region} + (1-\alpha) F_{Boundary} \qquad (1)$$

This unified framework aims at obtaining a robust *snake* evolution less sensitive to noise. The block diagram of the implemented system is presented in Fig. 7. The segmentation process starts with a filtering step. Then a model of the image based on the luminance histogram is used to calculate the external forces driving the surface evolution. Finally one surface is evolved for every object in the image and included in the resulting mask.

Assuming that the image is composed of a set of luminance homogeneous components, a statistical image model is generated from the histogram. Under this assumption the histogram can be decomposed as a mixture of Gaussians,

$$\begin{cases} p(x) = \sum_{k=1}^{N} a_k g_k(x) \\ g_k(x) = \dfrac{1}{\sqrt{2\pi}\,\sigma_k} e^{-\frac{(x-m_k)^2}{2\sigma_k^2}}, k = 1,\ldots,N \end{cases} \qquad (2)$$

The Minimum Description Length (MDL) principle is used to automatically estimate the number of histogram classes (N) while the Maximum Likelihood estimation provides the Gaussian parameters $\{(a_k, m_k, \sigma_k), k=1,2\ldots N\}$, see equation (2). A cost function is iteratively computed combining in the same expression the likelihood (ℓ_i) of the model and the number of classes (G_i) needed for the mixture. In order not to over-segment the image the process stops when the first minima of the cost function is reached. Expression (3) includes the weighting coefficients α and β,

$$MDL_i = \alpha \log(\ell_i) + \beta G_i, i = 1\ldots N \qquad (3)$$

Given an image model, the forces driving the surface evolution are calculated. Since *snake* propagation subject to the image gradient is very noise sensitive, a region force (F_{Region}), which corresponds to the object included in the evolving surface, has also been included in the PDE. This force can be directly derived from the statistical image model. Thus for a pixel s the associated region force is as follows,

$$F_{\text{Region}}(k) = \log\left(\frac{P_{R_{outside}}(I(s))}{P_{R_{inside}}(I(s))}\right)|\nabla j| \qquad (4)$$

where $P_{R_{outside}}$ and $P_{R_{inside}}$ are the outer and inner region probabilities for the object being segmented. When the pixel is assumed to be inside the object, the force tends

to shrink the surface, whereas the opposite behaviour appears when the pixel does not belong to the object.

Based on a probabilistic boundary estimation, a density function image is calculated where each pixel of this image represents the probability of that pixel being a boundary pixel.

Using the Gaussian mixture decomposition, every local neighborhood can be associated to a region model according to its luminance average. The Bayes theorem is used to derive the expression in (5) from the boundary probability (B_i) given a pixel luminance neighborhood ($I(N(s))$).

$$p(B_i \mid I(N(s))) = \frac{p(I(N(s)) \mid B_i)}{p(I(N(s)) \mid B_i) + p(I(N(s)) \mid \overline{B}_i)} P(B_i) \qquad (5)$$

The boundary-based force is presented in equation (6). For every object the boundary density distribution can be represented as an image. Figs. 8(a), 8(b) and 8(c) show the resulting boundary distribution images calculated from the original image shown in Fig. 8(d), whose histogram has been modeled with $N=3$ classes.

$$F_{\text{Boundary}}(k) = \left(g(p_{B_i}(s), s_{B_i})k + \nabla g(p_{B_i}(s), s_{B_i}) \cdot \frac{\nabla j}{|\nabla j|} \right) \nabla j \qquad (6)$$

For every object, a different surface is initialized and evolves. The surface evolution is modeled by a PDE which includes the region and the boundary information terms.

3.2.2. Object tracking

The above framework presented for still image segmentation can be extended to video sequences. Since level-sets deal with topological changes in the objects, a straightforward video segmentation can be implemented by projecting the final surface obtained for a given frame into the next frame. After re-initialization, the surface will naturally evolve driven by the forces computed for the present frame. Assuming small object motion, the number of iterations can be radically decreased speeding up the segmentation process while maintaining accuracy in the results.

3.2.3. Results

Fig. 8(e) shows the final image partition with $N=3$ regions. It is important to notice that, while regions belonging to the same object correspond to a single volume inside the evolving surface, they might not be connected as single regions at the zero level-set. Therefore, this level-set implementation allows topological changes for the segmented object.

4. CONCLUSION AND FUTURE WORK

In the short term, work in the Qimera project will focus on an evaluation of the analysis techniques reported here. In this context, we will consider criteria such as accuracy (e.g. contour localisation), under/over segmentation, temporal coherency and even factors such as module execution speed. We will evaluate segmentation results against the ground truth segmentations available for commonly used test sequences. For objective evaluation metrics, we will investigate the use of those proposed within the Cost 211 project [10] and explore the actual usefulness of these metrics. We also explore other existing metrics and develop our own metrics where appropriate.

We will also continue to work towards a longer-term research goal of collaboratively developing a set of *Inference Engines* [1]. The term Inference Engine (IE) is given to the module within the Qimera system that will eventually combine the results of a number of different analysis techniques in order to produce a final object segmentation. It is envisaged that a number of different IEs, targeting different types of segmentation problems, will be developed and evaluated.

5. ACKNOWLEDGEMENT

This material is based upon work supported by the IST programme of the EU in the project IST-2000-32795 SCHEMA. The support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged.

6. REFERENCES

[1] N. O'Connor, T. Adamek, S. Sav, N. Murphy, S. Marlow, "Qimera: a software platform for video object segmentation and tracking", WIAMIS 2003, London, April 2003, pp 204-209.

[2] E. Tuncel, L. Onural, "Utilization of the recursive shortest spanning tree algorithm for video-object segmentation by 2-D affine motion modelling", IEEE Transactions on Circuits and Systems for Video Technology, vol. 10, no.5, August 2000

[3] I. Kompatsiaris and M. G. Strintzis, "Spatiotemporal Segmentation and Tracking of Objects for Visualization of Videoconference Image Sequences", IEEE Trans. on Circuits and Systems for Video Technology, vol. 10, no. 8, Dec. 2000.

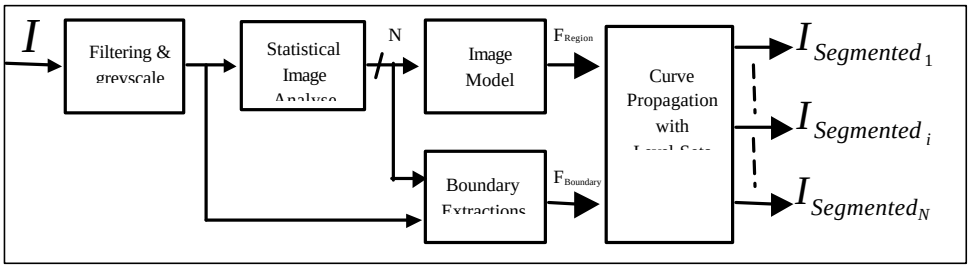
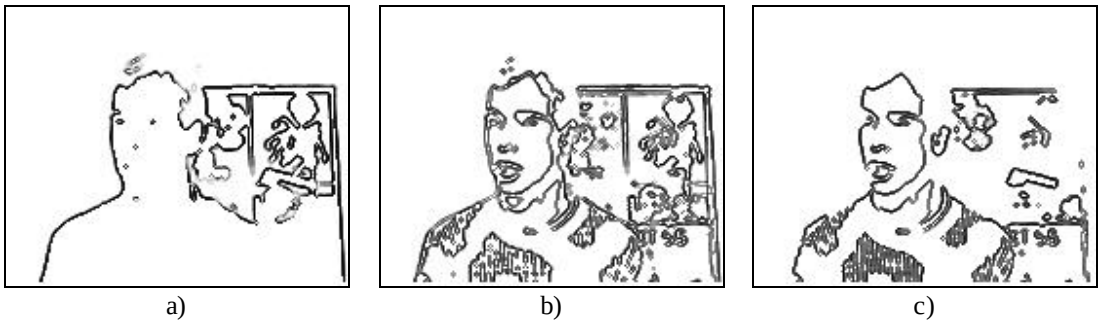


Figure 7: Block diagram of the *snake* segmentation module.



a) b) c)

Figure 8: Images a, b and c are probability density functions of object boundaries with N= 3. In the lower row, we show the original input image (d) and the resulting segmentation mask (e).

d) e)

[4] N. V. Boulgouris, I. Kompatsiaris, V. Mezaris, D. Simitopoulos and M. G. Strintzis, "Segmentation and Content-based Watermarking for Color Image and Image Region Indexing and Retrieval", EURASIP Journal on Applied Signal Processing, April 2002.

[5] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image Segmentation using Expectation-Maximization and its Application to Image Querying", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 24, No. 8, pp. 1026-1038 Aug. 2002.

[6] G. Schwarz, "Estimating the Dimensions of a Model", Annals of Statistics, Vol. 6, pp. 461-464, 1978.

[7] J. Rissanen, "Modeling by Shortest Data Description", *Automatica*, vol. 14, pp. 465-471, 1978.

[8] N. O'Connor, S. Marlow, "Supervised semantic object segmentation and tracking via EM-based estimation of mixture density parameters", Proceedings NMBIA'98 (Springer-Verlag), pp. 121-126, Glasgow, July 1998.

[9] N. Paragios and R. Deriche, "Coupled geodesic active regions for image segmentation: A level set approach", In European Conference in Computer Vision, Dublin, Ireland, June 2000.

[10] P. Villegas, X. Marichal, A. Salcedo "Objective Evaluation of Segmentation Masks in Video Sequences", WIAMIS'99, Berlin, May 1999, pp. 85-88.